

## BILLING REDUNDANT ATM CONNECTIONS

### FIELD OF THE INVENTION

The present invention is related to billing redundant ATM connections. More specifically, the present invention is related to billing redundant ATM connections based on an accurate account based on the total cell counts received, transmitted or dropped obtained from both the primary and the secondary components.

### BACKGROUND OF THE INVENTION

ATM switches can be used to setup connections between logical interfaces on the switch. These connections can allow data to flow through the switch in small pre defined packets called atm cells. The switch hardware can be configured to count the number of cells that have been received, transmitted or dropped through the switch. These cell counts are then utilized by various billing software packages to bill users for the amount of traffic that they have sent through the switch.

In case of redundant switch architectures a hardware component (primary component) is backed up by another identical hardware component (secondary component). When the primary component fails, the secondary component takes over. Failure can be defined as physical hardware component failures or as loss of signal/carrier on a fiber carrying data.

The secondary component taking over from the primary component prevents existing connections that have been setup from being torn down. Minimal data loss is experienced during this transition. However for redundant switch architectures cell flows through the primary and secondary components will be different depending on the component that is active. Therefore cell counts

collected from the primary and secondary components will be different.

For billing connections that are setup in redundant hardware components, an accurate account of the total cell counts received, transmitted or dropped needs to be obtained from both the primary and secondary components and then combined in software appropriately.

This invention solves the problem of collecting the relevant hardware cell counters, combining them in software and ensuring that the cell counts collected represents an accurate count of total cells received, transmitted or dropped through the ATM switch.

#### SUMMARY OF THE INVENTION

The present invention pertains to an apparatus for transferring data on a network. The apparatus comprises a switch comprising a primary component for switching the data through the switch. The apparatus comprises a secondary component for switching the data through the switch if the primary component fails. The apparatus comprises a mechanism for counting the data that has been received, transmitted or dropped by the switch without including any redundancy in the counting of the data due to the primary component and secondary component both able to switch data through the switch.

The present invention pertains to a method for transferring ATM cells on a network. The method comprises the steps of switching the ATM cells with an ATM switch of the ATM network having a primary component for switching the cells and a secondary component for switching the cells if the primary

component fails. Then there is the step of counting the cells that have been received, transmitted or dropped by the switch without including any redundancy in the counting if the cells due to the primary component and secondary component both able to switch the  
5 cells through the switch.

#### BRIEF DESCRIPTION OF THE DRAWINGS

In the accompanying drawings, the preferred embodiment of the invention and preferred methods of practicing the invention are illustrated in which:

10 Figure 1 is a schematic representation of how the counter-combining algorithm of the present invention operates.

Figure 2 is a schematic representation of where cell counts are obtained from in the present invention operates.

15 Figure 3 is a schematic representation of the physical interfaces which count the transmitted cells, received cells or dropped cells.

Figure 4 is a schematic representation of the apparatus of the present invention.

#### DETAILED DESCRIPTION

20 Referring now to the drawings wherein like reference numerals refer to similar or identical parts throughout the several views, and more specifically to figure 4 thereof, there is shown an apparatus 10 for transferring data on a network 12. The apparatus 10 comprises a switch 14 comprising a primary component 16 for  
25 switching the data through the switch 14. The apparatus 10

comprises a secondary component 18 for switching the data through the switch 14 if the primary component 16 fails. The apparatus 10 comprises a mechanism for counting the data that has been received, transmitted or dropped by the switch 14 without including any  
5 redundancy in the counting of the data due to the primary component 16 and secondary component 18 both able to switch 14 data through the switch 14.

Preferably, the counting mechanism 20 counts data as ATM cells from the network 12 which is an ATM network 12. The counting  
10 mechanism 20 preferably includes a cell recording module 22 which combines cell counts calculated from the primary and secondary components 16, 18. Preferably, the module collects cell counts for each connection through the switch 14.

The module preferably counts cell counts from either the  
15 primary component 16 or the secondary component 18 depending upon which one is actually having the connection through the switch 14, otherwise known as an active component. Preferably, the primary component 16 and the secondary component 18 each have physical interfaces 24 and a logical interface 26 for both components that  
20 maintains a switchover count which counts whenever a physical component takes over the connection because the other component has failed, otherwise called a switchover event, and each physical interface 24 counts the number of cells of the connection that have passed through it.

25 The logical interface 26 preferably increases its switchover count whenever it receives a switchover event and clears cell counts on the active component. Preferably, the module counts the cell counts of the logical interface 26 on the active component for each connection in predetermined intervals.

The module preferably reviews the logical interface 26 at each interval to determine if the component with which it is associated has its switchover count increased since a previous interval. Preferably, the module counts cell counts from each physical interface 24 to determine a total number of cells that has passed through the switch 14, unless the switchover count has increased since the previous interval, in which case the cell count from the previous interval is subtracted from the cell count from the logical interface 26.

The present invention pertains to a method for transferring ATM cells on a network 12. The method comprises the steps of switching the ATM cells with an ATM switch 14 of the ATM network 12 having a primary component 16 for switching the cells and a secondary component 18 for switching the cells if the primary component 16 fails. Then there is the step of counting the cells that have been received, transmitted or dropped by the switch 14 without including any redundancy in the counting if the cells due to the primary component 16 and secondary component 18 both able to switch 14 the cells through the switch 14.

Preferably, the counting step includes the step of counting the cells with a cell recording module 22 which combines cell counts from the primary and secondary components 16, 18. The counting step preferably includes the step of counting cells with the module for each connection that passes through the switch 14. Preferably, the counting step includes the step of counting with the module from either the primary component 16 or the secondary component 18 depending on which one is actually having the connection through the switch 14, otherwise known as an active component.

The counting step preferably includes the step of maintaining a switchover count in each logical interface 26 of each component which maintains a switchover count which counts whenever the associated components takes over the connection because the  
5 other component has failed, otherwise called a switchover event, and counting with each logical interface 26 the number of cells of the connection that has passed through it. Preferably, the maintaining step includes the step of increasing by each logical interface 26 its switchover count whenever it receives a switchover  
10 event and then clearing cell counts on the active component.

The counting step preferably includes the step of counting the cell counts of each logical interface 26 on the active component for each connection at predetermined intervals. Preferably, the counting step includes the step of reviewing each  
15 logical interface 26 at each interval to determine if the component with which it is associated has its switchover count increased since a previous interval.

In the operation of the invention, the call recording module is responsible for collecting cell counts from various per  
20 connection hardware counters. This module implements the software algorithm for combining cell counts collected from the primary and secondary components 16, 18.

The hardware component that is actively involved in switching cells and carrying data is referred to as the "active"  
25 component. During the time that the switch 14 is running, the active component can be either the primary or secondary component. When the primary component 16 is running, the active component is the primary component 16. When the primary fails and the secondary takes over, the active component is the secondary component 18. If  
30 the secondary fails and the primary takes over again, the active

component will be the primary component 16. The switchover between the primary and secondary components 16, 18 can occur multiple number of times while the switch 14 is active.

Every time a switchover occurs (i.e. the primary hardware fails and the secondary takes over or vice versa), a switchover event is generated on all logical interfaces 26 that are associated with the hardware component that switched over. Each logical interface 26 maintains a switchover count. On receiving a switchover event the logical interface 26 increases its switchover count and clears the cell counts on the active hardware component.

The following algorithm describes the set of actions taking place when a logical interface 26 receives a switchover event.

Increment logical\_interface->switchover\_count;

For all connections going through this logical interface {

Zero hardware counts on the "active interface"

}

The call recording module keeps track of cell counters on the "active" hardware component by checking them on user-configured intervals. At every user-configured interval, the logical interfaces 26 associated with each connection are looked at to identify if the hardware associated with those interfaces switched over. If the hardware switched over (identified by the increased switchover count) between the previous interval and this one, cells counts are combined in one way. If the hardware did not switch





```
} else {  
  
    total_cells += last_interval_cells;  
  
    last_interval_cells = current cells read from hardware - cur_hw_ctr;  
  
    cur_hw_ctr = current cells read from hardware;  
  
5      }  
  
}
```

Figure 1 and Table 1 illustrate how the Counter-combining Algorithm works. Figure 1 assumes that cells are uniformly sent at about 100 cells/minute.

10 Notation:

Time : User Configured Interval when the hardware cell counts are collected. In this example the User Configured Interval is set to 60 seconds.

15 Actual Cell Count : Cell Counts obtained from the current hardware component.

Standby | Active H/W : Cell Counts from Standby/Active hardware component after Ctrs after Switchover switchover. (The terms "standby" and "active" refer to the standby and active hardware components after switchover.

20 Delayed Cell Count : Cell counts after discarding 1 interval of data

Interval Time	Connection created	0	1	2	3	4	5	6	7	8	9	10
Total Cells	0	0	0	100	200	300	300	350	350	400	400	425
Last Interval Cells	0	0	100	100	100	100	50	100	50	100	25	100
Current HW Ctr	0	0	100	200	300	400	50	150	50	150	25	125
Call Recording Switchover Ctr	5	5	5	5	5	5	5	6	6	7	7	9
Logical Interface Switchover ctr	5	5	5	5	5	5	6	6	7	7	9	9
Delayed Cell Count [After Losing 1 Interval]	0	0	0	100	200	300	300	350	350	400	400	425
Physical Hardware Ctr	0	0	100	200	300	400	50	150	50	150	25	125

**Table 1**

Total Cells, Last Interval Cells, Current HW Ctr and Call recording Switchover Ctr are per connection structures stored in the call recording database.

Logical Interface Switchover Counter is stored in the logical interface switchover counter and is incremented when ever a switchover occurs.

The Delayed Cell Count is manually calculated by looking at cell counts after switchover occurs. This cell count is obtained by discarding data obtained in an interval where one or more switchovers occur and by adding all other cell count data after a delay of one interval. This count should be similar to the Total

Cells count arrived at by using the algorithm.

Physical Hardware ctr are the cell counts obtained by looking at the actual "active" hardware counters.

Figures 2 and 3 show where the cell counts are obtained from. Figure 3 shows the physical interface 24 which counts the transmitted cells, and the physical interface 24 which counts the received cells and the dropped cells.

